HOW FAR SHOULD I LOOK? A NEURAL ARCHITECTURE SEARCH STRATEGY FOR SEMANTIC SEGMENTATION OF REMOTE SENSING IMAGES

M. C. M. de Paulo[a,*], J. N. Turnes[c], P. N. Happ[b], M. P. Ferreira[a], H. A. Marques[a], R. Q. Feitosa[b]

[a] Dept. of Defense Engineering, Military Institute of Engineering, Rio de Janeiro, Brazil - (mauricio.paulo, matheus, haroldomarques)@ime.eb.br
[b] Dept. of Electrical Engineering, Pontifical Catholic University of Rio de Janeiro, Rio de Janeiro, Brazil - patrick.happ@gmail.com, raul@ele.puc-rio.br
[c] Dept. of System Design Engineering, University of Waterloo, Waterloo, Canada - jnoaturn@uwaterloo.ca

Commission III, WG III/1

KEY WORDS: Neural Architecture Search, Semantic Segmentation, Remote Sensing, Satellite imagery, Convolutional Neural Networks

ABSTRACT:

Neural architecture search (NAS) is a subset of automated machine learning that tries to find the best neural network to perform a given task. In this article, a network search space is defined and applied to perform the semantic segmentation of satellite imagery. Due to the spatial nature of the data, the search space uses cells that group parallel operations with kernels of different sizes, providing options to accommodate the neighborhood information required to perform a better classification. The architecture search space follows a UNet-like network. The proposed approach uses scaled sigmoid gates, a strategy for network pruning that was adapted to search for the best operations on the cell search space. The architecture achieved by the proposed approach uses wider kernels on lower resolution feature maps, which leads to the interpretation that some pixels required information from pixels farther away than expected. The resulting network was compared to a very similar UNet-like network that only used 3x3 convolutions. The resulting network shows slightly better results on the test set.

1. INTRODUCTION

Satellite imagery is used in many research fields to describe and understand the spatial and temporal evolution of the Earth's surface. Cartography has a particular interest in these images because of the growing availability of satellite imagery that can be used to update databases and maps. These high-resolution images provide worldwide coverage periodically.

Machine learning can be used to extract information from satellite imagery. For example, semantic image segmentation concepts can be used with deep neural network (DNN) methods to apply labels of land cover to every pixel in a satellite image.

Neural networks require a large number of training samples, thus benchmark datasets such as Semcity Toulouse (Roscher et al., 2020) are of great value. The growing availability of labeled datasets and the improvements in graphic processing units (GPU) are enabling further research.

A Convolutional Neural Network (CNN) uses convolution layers instead of dense layers, such as the multilayer perceptrons used on earlier neural network research. The research on CNN reached much better accuracy in semantic segmentation than previous machine learning approaches so far. One of the main difficulties in improving the results of a CNN is finding which architecture suits a given task. For example, the number of layers, the type of transformation applied on each layer, how they are connected and the number of convolution filters are all designed by the researcher. A few popular architectures have been employed for specific tasks, while creating new, faster, and more accurate architectures is also a research topic.

* Corresponding author

Neural Architecture Search (NAS) is a subset of Automated Machine Learning that tries to automate some of the decision-making processes on finding the best architecture to solve a problem. (He et al., 2021). A brute force comparison of many network configurations could find the best architecture given a few options, but training semantic segmentation networks is computationally intensive and there are countless possibilities. Thus, specific algorithms for NAS have been investigated (Liu et al., 2019b; Liang et al., 2019). Neural architecture learning (NAL) has a similar goal, but instead of searching among every possible network, the algorithms seek to optimize the network given a set of options designed as the search space (Guo et al., 2021). Most of the NAS algorithms are trained and tested to solve computer vision problems (Liu et al., 2019b; Liang et al., 2019; Peng et al., 2020) and a few have been adapted to address the semantic segmentation task (Liu et al., 2019a; Peng et al., 2020).

This paper describes the application of a NAL algorithm for the task of semantic segmentation of multispectral satellite images. The proposed approach uses a previously researched scaled sigmoid layer that can be used to prune a neural network (Guo et al., 2021). It was originally presented as a tool to prune layer channels, but we tested it for pruning network paths. The search algorithm uses the stochastic gradient descent (SGD) to find a subset of the operations on the cell search space that optimizes accuracy. A UNet-like (Ronneberger et al., 2015) network was defined as the structure for the network search space, with different convolution kernel sizes as options on the cell search space. The results comparing the network designed by the algorithm with a regular UNet that uses 3x3 kernels are presented and discussed. Due to spatial nature of the data, the size of the convolution filters that remained on the network provide an in-

tuitive interpretation of how far (in meters) the network has to look to optimize the classification of a pixel.

2. RELATED WORK

2.1 Deep Learning for image segmentation

There are several works available on the literature that indicate that the encoder-decoder architecture is a reasonable and flexible option for satellite image semantic segmentation (Neupane et al., 2021; **?**). The UNet is an architecture that inspired many fully convolutional network (FCN) for semantic segmentation. The main concept behind the UNet is the use of skip connections via concatenation from encoder layers to decoder layers (Ronneberger et al., 2015). This allows reusing features with different receptive fields and prevents the vanishing gradients problem. Figure 1 illustrates the general architecture used on encoder-decoder networks similar to UNet.

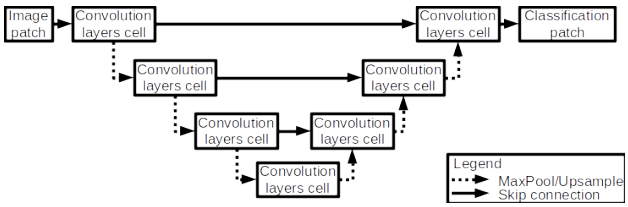

Figure 1. UNet-like general architecture.

The encoder part of UNet-like networks (layers on the left side of Figure 1) is used to extract features that describe decreasingly lower levels of details. Each "Max pool" layer creates a lower resolution version of the data processed through earlier convolution layers. The right part of the network is the decoder part. It uses up sampling, thus increasing the level of details until the original image size is reached. The skip connections procedure connects the same level layers using a concatenation operation. This way, the network brings the features extracted on each level back before the next set of convolutions. In doing so, some features that were lost on the lower resolutions can be reintegrated for later use on the classification. (Ronneberger et al., 2015)

2.2 Differentiable Neural Architecture Search (DNAS)

Automated machine learning (AutoML) is a research field that focuses on building systems without human intervention (He et al., 2021). Concerning neural networks, there are many approaches of how to address the network optimization problem and each has its own distinct research field. Among the most popular, there are evolutionary algorithms, grid and random search, reinforcement learning and gradient descent (He et al., 2021).

The use of the SGD to learn the architecture of a network is called Differential Neural Architecture Search, which was first proposed by (Liu et al., 2019b) with an algorithm named Differentiable Architecture Search (DARTS). The main advantage of this algorithm was the reduced training cost. While previous approaches required hundreds or thousands of days to reach the final architecture, DARTS required a few days on CIFAR-10 dataset. On the other hand, DARTS has all the disadvantages of SGD, such as finding non-optimal solutions depending on the training data and search space architecture (Liu et al., 2019b).

Figure 2 presents a CNN as an acyclic graph that uses connected layers, beginning on the input data and ending on the output data. These layers represent operations on the data received from the previous layer.

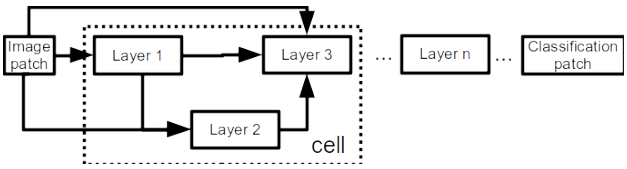

Figure 2. A general illustration of a CNN represented as an acyclic graph.

NAS algorithms work on finding a subset of these infinite options. The literature provides several options for the operations that can be performed as a layer, so most NAS algorithms reduce those options to a finite set that is tested. In DARTS, for example, the operations chosen were: 3×3 separable convolutions, 5×5 separable convolutions, 3×3 dilated separable convolutions, 5×5 dilated separable convolutions, 3×3 max pooling, 3×3 average pooling, identity and zero (Liu et al., 2019b). The set of operations may differ depending on the implementation and this can be a relevant factor in network performance. The operations chosen by Liu et al. (2019b) were tested for whole scene image classification.

DARTS uses the concept of cells and nodes, using a search space similar to NASNet (Liu et al., 2019b). The cells represent a group of layers and it's connections. The network is defined as a sequence of repeated cell (Zoph et al., 2018). All these cells have the same architecture and share the same weights. In order to find which operations are present on each cell, DARTS connects every layer option in parallel. A node is the feature map representing the sum of these layers. Figure 3 illustrates DARTS search with 4 nodes (0,1,2,3) and 3 operations (represented by the colored lines). Every node is connected to every previous node with each operation. DARTS gives a weight α, related to the architecture, to each operation and the operations have weights w. α represents the relative weight of each operation when compared to the other options. The training uses SGD in two steps: weights (w) and architecture (α). When the weights (w) are the trained, the architecture weights (α) are frozen, thus the network is pushed to a better representation of the samples. During architecture training, α is trained and w is frozen, changing the influence of each operation as a whole. In the end, α of the operations that arrive in a given node are compared using a softmax.

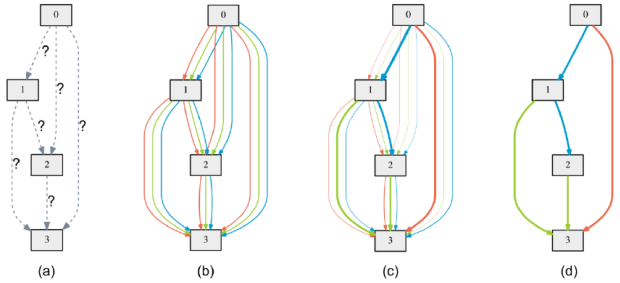

Figure 3. DARTS search strategy: Begins with every operation, then excludes the ones with low probability. Source: (Liu et al., 2019b)

The search described in Figure 3 is only possible because of the continuous relaxation of the operations search. In other words, the choice of which operation to use is binary, but the weighted sum of the operations results is continuous, thus the gradient can be used to search for the best option. Other researchers propose different continuous relaxation strategies (Guo et al., 2021; Kim et al., 2019) because the softmax strategy requires building and training a network with every possible operation. This is a key concept on NAS because the results of the searched network are affected directly by the chosen operations.

A recently developed strategy uses the scaled sigmoid (SS) layers (Guo et al., 2021). In this approach, every operation in the search space is followed by a SS layer. The SS layer is trained during the architecture training step (similar to α in DARTS), while every other layer is trained during weights training (w). This strategy provides flexibility because the network search space can be designed to improve specific parts of a given network. In addition, the SS layers are also applied in the channel wise weights (Guo et al., 2021). This means that the SS layer can be used to optimize the number of filters in a given convolution layer, improving performance with a smaller network.

2.3 NAS on semantic segmentation

Despite the popularity of NAS research, many important articles such as Liu et al. (2019b), Jin et al. (2019) and Liang et al. (2019) do not approach the semantic segmentation problem. Among the articles that target semantic segmentation, Auto-DeepLab (Liu et al., 2019a) used the architecture search from DARTS and introduced a cell-based search using cells similar to DeepLab (Chen et al., 2018).

There is also some work on medical image segmentation (Kim et al., 2019), where the authors proposed the use of an architecture search space similar to a UNet. The proposed algorithm optimized the choices and connections of layers on 3 different cells (reduction, expansion and normal) on a UNet-like architecture. Additionally, they used a Gumbel-softmax to perform continuous relaxation. Regarding NAS for semantic segmentation of satellite images, there are few articles in the literature (Zhang et al., 2020; Peng et al., 2020), that uses sequences of repeated cells, similar to DARTS (Liang et al., 2019).

3. PROPOSED APPROACH

This article aims to use DNAS algorithms to improve the design of UNet-like network by replacing some 3x3 convolution layers with parallel layers of different kernel sizes, then pruning the network using SS layers. Repeatable cells, that represent a sub-network, might not be well suited for this purpose, because it would force the network to use the same layers despite how many MaxPool layers were applied before it. In this research, the constraint of a single repeatable cell was lifted, allowing the cells to be different. Because of that, the memory consumption of the search was increased, so the architecture search space was narrowed. Figure 4 illustrates the architecture search space used in this research. It followed a UNet-like construction, similar to Kim et al. (2019), but there are 3 encoder cells (with white background) that were searched using the SS layers (Guo et al., 2021).

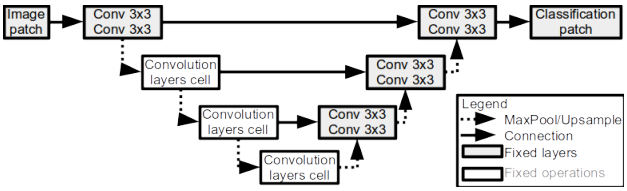

Figure 4. Architecture search space in this research.

Each of the "Convolution layers cells" represent a group of layers. To address the kernel size search that was the main idea on this research, each cell contained a set of dilated convolution layers of different kernel sizes. The search algorithm was used to choose which kernel size would better fit the training data. Every cell in gray in Figure 4 was not part of the search space, so these were not changed by the search algorithm.

To perform the search, the SS layers from Guo et al. (2021) were added to the network after every optional layer. Equation 1 represents how the sigmoid of a given layer is computed. δ_i is a growing constant that increases exponentially with each epoch i. s_l represents the weights of the SS layer. In this research, a single weight was applied to every channel. The result of applying an SS layer l to a feature map x is defined as the function $f_l(x)$ in Equation 2.

$$sigmoid(\delta_i.s_l) = 1/(1 + e^{-\delta_i.s_l}) \quad (1)$$

$$f_l(x) = x \cdot sigmoid(\delta_i.s_l) \quad (2)$$

Each SS begins with $s_l = 0$, thus the SS layer multiplies the inputs by 0.5. When the architecture training begins, the gradient pushes s_l to optimize the network. Since every layer is frozen, except the SS layers, only the s_l weights are going to be trained to improve the network classification result. When s_l grows, the SS layer multiplier gets near 1 and when s_l lowers, the SS layer multiplier approaches 0. When every SS layer reaches 0 or 1, the network can be pruned to remove the layers with $sigmoid(\delta_i.s_l) = 0$.

Figure 5 represents the cell search space. The first convolution layer (in gray) has a fixed 3x3 kernel. The following layers represent the operations with different kernel sizes: convolution 1x1, convolution 3x3, convolution 3x3 with dilation 2 and convolution 3x3 with dilation 3.

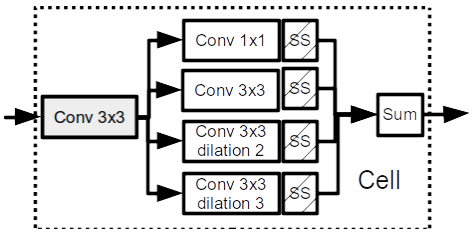

Figure 5. Cell search space on this research.

The training was performed using the alternated search, with two steps of weight training followed by one step of architecture training. During the weight training steps, every layer except the SS layers were trained. During the architecture training steps, every layer was frozen and only the SS layers were

trained. The training ended when the sigmoid values of each SS layer reached either 0 or 1, considering a threshold of 0.001. This state means that the network has reached a binary indication of which layers should be used. The sigmoid values for each layer were concatenated in an architecture vector, similar to DARTS.

Following Guo et al. (2021), a different loss function was used during the architecture an the weight training. To evaluate the semantic segmentation problem, the categorical cross-entropy was used as the base loss function (L_0). Since each SS layer has a single weight s_l for the entire layer, the modified loss was defined as Equation 3. The factor δ_i was defined as 2^i where i is the number of epochs, starting on 0. $\lambda = 5 \cdot 10^{-5}$ was used, following the best result found by Guo et al. (2021).

$$L = L_0 + \lambda \sum_{l=0}^{n_{\text{ss layers}}} sigmoid(\delta_i . s_l) \tag{3}$$

After the architecture search, the network was rebuilt using the architecture vector found. The network was trained using the same training and validation sets from the previous step and then evaluated on the test images. A similar UNet network, using only 3x3 convolutions instead of the search cell, was used as a baseline to evaluate if the network found by the algorithm is a better option than the classic architecture approach.

Every training step was performed using ADAM optimizer with a learning rate of 0.001, which is the default on the library used Abadi et al. (2016). Both the searched network and baseline network were trained for 100 epochs, with an early stopping mechanism with patience of 20 epochs, monitoring validation accuracy improvements of at least 0.1%. Both networks ended their training before the 100 epochs limit was reached. Tests were performed on Google Colab Pro cloud platform using, reportedly, NVIDIA P100 with 16GB of GPU memory.

4. DATA SET

International Society of Photogrammetry and Remote Sensing (ISPRS) Working Group II/6 has offered a new benchmark for semantic image segmentation: Semcity Toulouse (Roscher et al., 2020). This dataset contains 16 high-resolution images of spatial resolution of 0.5m, generated from the panchromatic fusion of 3 multispectral bands (Red, Green, Near-infrared). Currently, out of the 16, 4 images are labeled for semantic segmentation. Table 1 contains a list of the classes labeled on the dataset and the percentage of the total samples each represents.

class	% labeled samples
impervious surface	23%
building	23%
pervious surface	30%
high vegetation	16%
car	2%
water	3%
sport venues	3%

Table 1. Classes available on the Semcity Toulouse dataset

Figure 6 shows, on the left, the high-resolution multispectral images available on the Toulouse dataset and its respective classes, on the right. The images represent tiles of the original dataset (Roscher et al., 2020).

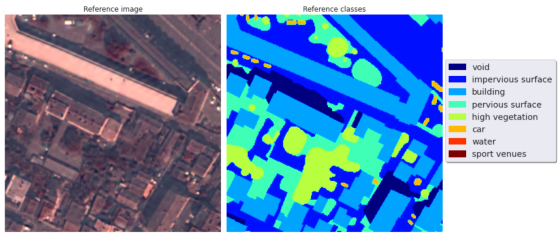

Figure 6. Example tile from the Toulouse dataset: image and labels.

Out of the 4 images, 3 were used for training (images 3,4 and 7) and one for testing (image 8). Using a stride of 224 pixels, the individual images were tiled in patches of 256x256 pixels with 8 channels (spectral bands) each. 675 patches were generated for training and 225 for testing. From the training area, 540 patches where use for training and 135 for validation.

5. RESULTS AND DISCUSSION

5.1 Architecture performance

The network configuration that represents the result of the architecture training with the SS layers will be addressed as NAS network. Figure 7 shows the accuracy and loss on both training and validation processes during each epoch of architecture training. The training ended when the architecture vector only had 0 and 1 values. The first epochs have worse accuracy, but architecture training only begins on the third epoch. This way, the starting point for the architecture training reached around 70% accuracy in the training set, which is better than the first epoch. The impact of the proportion was not fully tested, so the number of epochs for weights and architecture on each training cycle were implemented as a hyperparameter.

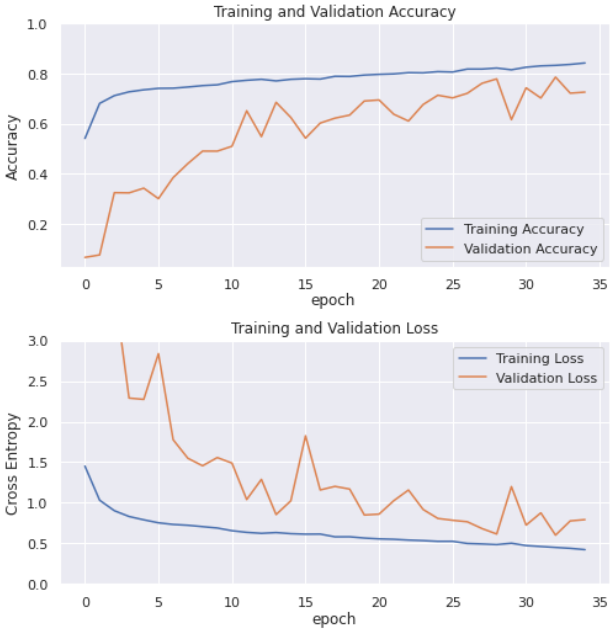

Figure 7. Accuracy on training and validation data during architecture training.

Figure 8 illustrates the NAS network. Every cell had one convolution 3x3 followed by the parallel convolution options. The term "Fixed 3x3" represents the initial convolution 3x3 of each cell. The layers bellow it represent the layers that remained on the NAS network. The gray boxes were not inside the search space and were not changed by the algorithm.

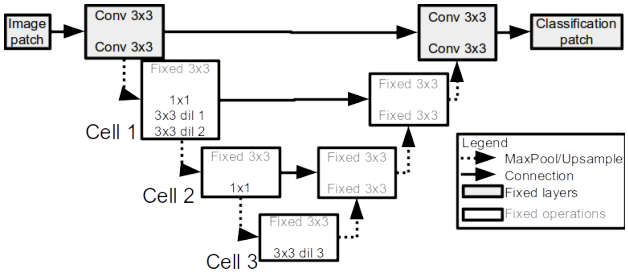

Figure 8. NAS network.

The NAS Cell 1 has a convolution of kernel 3x3 followed by 3 parallel convolutions (1x1, 3x3 dilation 1 and 3x3 dilation 2). This means that Cell 1 had every convolution, except the dilation 3. Differently, Cell 2 had only a convolution 3x3 followed by a convolution 1x1, which is the smallest possible. This brings an interpretation that the other kernels might not be contributing more than the previous dilated convolutions.

Cell 3 represents the highest level of the NAS network and the NAS cell has a convolution 3x3 followed by a dilated convolution 3x3 with rate 3, which is the largest possible kernel. Due to the number of previous MaxPools layers, this layer brings information that was not available on previous layers. For some pixels, that information might have been relevant. After three MaxPool layers, which represent a change in image size, the feature attribute on Cell 3 is only 32x32 pixels. This means that a kernel of 7x7 brings information from 12m away from the current pixel ($3pixels \cdot 2^3 MaxPools \cdot 0.5m$). This might have been relevant to classify pixels on open areas.

5.2 Classification performance

Table 2 presents the accuracy values achieved by each network on the different sample sets. The network used for architecture search has more parameters. While validation accuracy on both NAS and baseline networks was comparable, training accuracy was much higher on the NAS network and the accuracy obtained on the test set was slightly higher (2.03 percentage points). This is probably because the network architecture was decided using the training and validation data. It should be noted that the improvement in the classification accuracy was achieved with fewer parameters, which suggests that some of the convolutions regularly used on UNet-like networks are not contributing to improving results.

Network	Architecture search	NAS	Baseline
Parameters	1,916,628	1,123,592	1,213,320
Train. acc.	83.12%	93.67%	85.47%
Val. acc.	60.64%	80.56%	80.92%
Test acc.	-	76.12%	74.09%
Train. time	304s	542s	627s

Table 2. Summary of accuracy values and training time for the proposed and baseline approaches

The F1 score is the harmonic mean of precision and recall, described in Equations 4, 5 and 6 (Sokolova et al., 2006). Table 3 contains the F1 scores achieved after testing each network. Remarkably, the NAS network F1 score was higher or equal on every class. The average F1 shown on the last table row describes that overall higher F1 score.

$$precision = \frac{true\ positives}{true\ positives + false\ positives} \quad (4)$$

$$recall = \frac{true\ positives}{true\ positives + false\ negatives} \quad (5)$$

$$F_1 = \frac{2 \cdot precision \cdot recall}{precision + recall} \quad (6)$$

Class	NAS	Baseline
void	12.9%	4.6%
impervious surface	77.0%	74.6%
building	87.8%	87.8%
pervious surface	70.6%	67.2%
high vegetation	71.7%	68.8%
car	66.4%	64.2%
water	95.4%	95.2%
sport venues	68.0%	59.7%
average F1	68.7%	65.3%

Table 3. Summary of F1 scores achieved with each network

Figures 9 and 10 show the confusion matrix for the NAS and baseline networks respectively, using the same vertical scale. Comparing both matrices, it can be seen that the NAS network achieved worse results on impervious surfaces and water. A higher number of impervious surfaces pixels were classified as buildings and pervious. Also, a higher number of water pixels were classified as impervious surfaces. These differences might come from the wider look that the NAS network represents, because of increased kernel sizes.

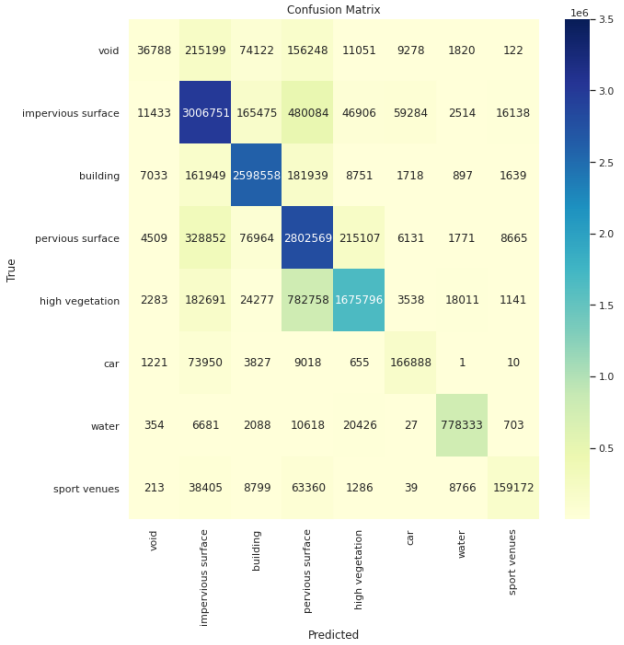

Figure 9. Confusion matrix of the NAS network.

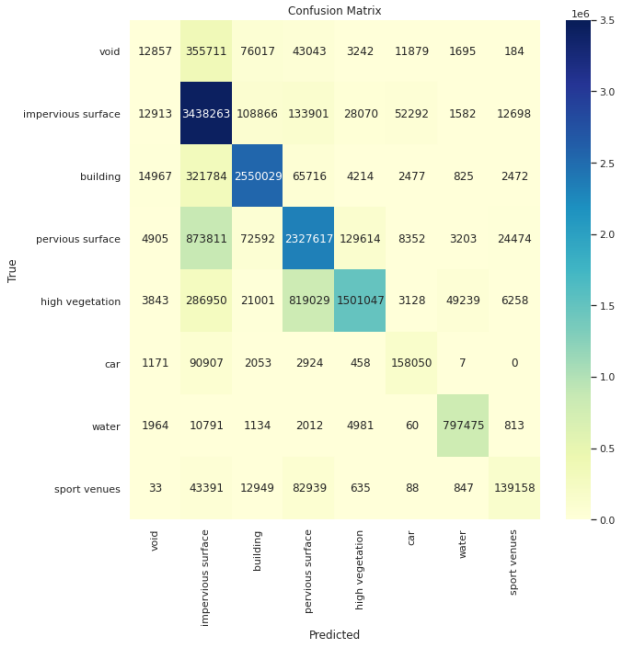

Figure 10. Confusion matrix of the baseline network.

Figure 11 provides a visual illustration of the classification results. The the improvement in the classification accuracy on the NAS network is very hard to evaluate visually. Both baseline and NAS networks have the same strengths and flaws as UNet-like networks, because of how the search space was designed. Some finer details were not well represented. For example, there were few water bodies and cars both on training and testing sets, thus both networks failed to identify two out of three water bodies. Building edges and the transitions between the high vegetation and pervious surfaces represent a large area of classification errors. The classification errors vegetation is unexpected, since the dataset uses infrared bands. There were fewer samples for the water bodies class, so there was a samples balancing issue.

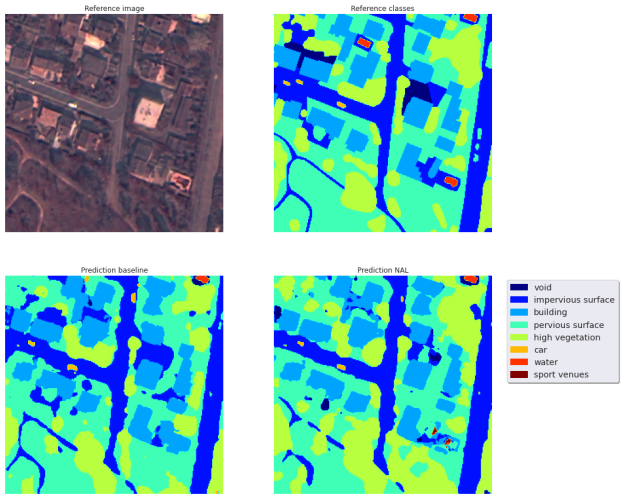

Figure 11. Visual results.

5.3 Lessons learned with the proposed approach

The SS layers strategy applied in this research is a very flexible tool that seems promising for future research. Guo et al. (2021) uses the SS layers to create more efficient networks from networks that had high performance already. Our experience with the SS layers indicate that there are more uses for this concept and that it can be integrated in different neural network researches. The channel-wise SS layers have an even broader research field, because it can be used to improve high performance networks that were consuming too much memory and possibly improve their accuracy.

A major issue with our search procedure was found during testing. When using the search cell on the decoder part of the network, the SS layers for all options on that cell could approach zero. This hindered the previous layers, even if they had already reached 1. To avoid this issue no cells were added to the decoder part of the network, which had fixed convolutions, and there are no sequential SS layers. The training step could force the architecture vector to save at least one operation among the parallel ones, but this was not tested.

Earlier tests were performed with separable convolutions, following the operations from (Liu et al., 2019b), and the results were compared to the baseline. Many changes to network structure and different kernel sizes were tested and, in every test, the baseline network reached better validation accuracy, while most networks reached higher training accuracy. These tests were discarded because the NAS network used separable convolutions, while the UNet used regular convolutions. After changing to regular dilated convolutions, the NAS network reached better accuracy values.

The output of a cell was tested using sum (Liu et al., 2019a) and concatenate (Liu et al., 2019b) layers without significant differences in the accuracy. There was an impressive increase in the number of parameters to evaluate because the number of output filters using concatenate layers is higher. Since GPU memory is a regular constraint, the sum operation was kept on the code.

6. CONCLUSION

The research described in this article is a very early experiment of how the scaled sigmoid layers can be used on satellite images for semantic segmentation. There are many state of the art architectures that were found through empirical tests that could be improved with SS layers, both searching for better layer options or searching for improvements in the number of channels on each layer.

The cell and architecture search spaces used in this article were found by trial and error. So despite being a search mechanism, the search space on NAS algorithms is still constrained by many decisions made by the user. NAS research still has a long path to reach the AutoML goal, thus this is a prominent research field. The searched network was chosen using a deep learning method based on training and validation data. Different images or different parameters may lead to a different network architecture.

The NAS network presented interesting aspects to the remote sensing community. The NAS network achieved higher accuracy with fewer parameters and used a wider kernel on the lower resolution part of the network. With a different search strategy, it might also solve the issues related to how many MaxPool layers are required for a given semantic segmentaion task.

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-V-3-2022-17-2022 | © Author(s) 2022. CC BY 4.0 License.

Further research is still required to bring the concepts integrated on this article to everyday use. The SS layer brought more flexibility to the search space, allowing researchers to design their own search space instead of using a previously defined one. We hope that this flexibility empowers new research on the field.

The code and dataset used on this research are available on Google Colab through the link `https://colab.research.google.com/drive/1a0Q4N3ftBL9iEm8G0mjQDoEHDnwavozQ` to foster future research.

7. ACKNOWLEDGEMENT

This work was supported by the Department of Science and Technology of the Brazilian Army. The study was performed during the postgraduate class ELE 2346 - Deep Learning, on to the Department of Electrical Engineering of the Pontifical Catholic University of Rio de Janeiro.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mane, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viegas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., Zheng, X., 2016. TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. *arXiv:1603.04467 [cs]*. http://arxiv.org/abs/1603.04467. arXiv: 1603.04467.

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A. L., 2018. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4), 834–848. Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence.

Guo, Q., Wu, X.-J., Kittler, J., Feng, Z., 2021. Differentiable Neural Architecture Learning for Efficient Neural Network Design. *arXiv:2103.02126 [cs]*. http://arxiv.org/abs/2103.02126. arXiv: 2103.02126.

He, X., Zhao, K., Chu, X., 2021. AutoML: A survey of the state-of-the-art. *Knowledge-Based Systems*, 212, 106622.

Jin, H., Song, Q., Hu, X., 2019. Auto-keras: An efficient neural architecture search system. *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 1946–1956.

Kim, S., Kim, I., Lim, S., Baek, W., Kim, C., Cho, H., Yoon, B., Kim, T., 2019. Scalable Neural Architecture Search for 3D Medical Image Segmentation. *arXiv:1906.05956 [cs, eess, stat]*. http://arxiv.org/abs/1906.05956. arXiv: 1906.05956.

Liang, H., Zhang, S., Sun, J., He, X., Huang, W., Zhuang, K., Li, Z., 2019. Darts+: Improved differentiable architecture search with early stopping. *arXiv preprint arXiv:1909.06035*.

Liu, C., Chen, L.-C., Schroff, F., Adam, H., Hua, W., Yuille, A., Fei-Fei, L., 2019a. Auto-DeepLab: Hierarchical Neural Architecture Search for Semantic Image Segmentation. *arXiv:1901.02985 [cs]*. http://arxiv.org/abs/1901.02985. arXiv: 1901.02985.

Liu, H., Simonyan, K., Yang, Y., 2019b. DARTS: Differentiable Architecture Search. *arXiv:1806.09055 [cs, stat]*. http://arxiv.org/abs/1806.09055. arXiv: 1806.09055.

Neupane, B., Horanont, T., Aryal, J., 2021. Deep Learning-Based Semantic Segmentation of Urban Features in Satellite Images: A Review and Meta-Analysis. *Remote Sensing*, 13(4), 808. https://www.mdpi.com/2072-4292/13/4/808. Number: 4 Publisher: Multidisciplinary Digital Publishing Institute.

Peng, C., Li, Y., Jiao, L., Shang, R., 2020. Efficient Convolutional Neural Architecture Search for Remote Sensing Image Scene Classification. *IEEE Transactions on Geoscience and Remote Sensing*, 1–14. https://ieeexplore.ieee.org/document/9194271/.

Ronneberger, O., Fischer, P., Brox, T., 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. *arXiv:1505.04597 [cs]*. http://arxiv.org/abs/1505.04597. arXiv: 1505.04597.

Roscher, R., Volpi, M., Mallet, C., Drees, L., Wegner, J. D., 2020. SemCity Toulouse: A Benchmark for Building Instance Segmentation in Satellite Images. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, V-5-2020, 109–116. https://www.isprs-ann-photogramm-remote-sens-spatial-inf-sci.net/V-5-2020/109/2020/.

Sokolova, M., Japkowicz, N., Szpakowicz, S., 2006. Beyond Accuracy, F-Score and ROC: A Family of Discriminant Measures for Performance Evaluation. D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, A. Sattar, B.-h. Kang (eds), *AI 2006: Advances in Artificial Intelligence*, 4304, Springer Berlin Heidelberg, Berlin, Heidelberg, 1015–1021. Series Title: Lecture Notes in Computer Science.

Zhang, M., Jing, W., Lin, J., Fang, N., Wei, W., Wozniak, M., Damaševičius, R., 2020. NAS-HRIS: automatic design and architecture search of neural network for semantic segmentation in remote sensing images. https://www.vdu.lt/cris/handle/20.500.12259/111012. Accepted: 2020-11-01T19:07:01Z.

Zoph, B., Vasudevan, V., Shlens, J., Le, Q. V., 2018. Learning Transferable Architectures for Scalable Image Recognition. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Salt Lake City, UT, 8697–8710.